



Education

# **Energy Efficiency Strategies for Storage in HPC Environments**

Alan G.Yoder, Ph.D., NetApp

## ➤ Energy Efficiency Strategies for Storage in HPC Environments

- ◆ With some large-scale HPC data center owners admitting that storage and computing resources can cost more to power over their lifetimes than to purchase, attention to energy management in HPC is timely. This talk will focus on storage; it will survey various storage schemes for HPC, ranging from Hadoop clusters to enterprise storage arrays, and compare energy usage and management in the various schemata. The potential contribution of point technologies such as data deduplication will also be covered.

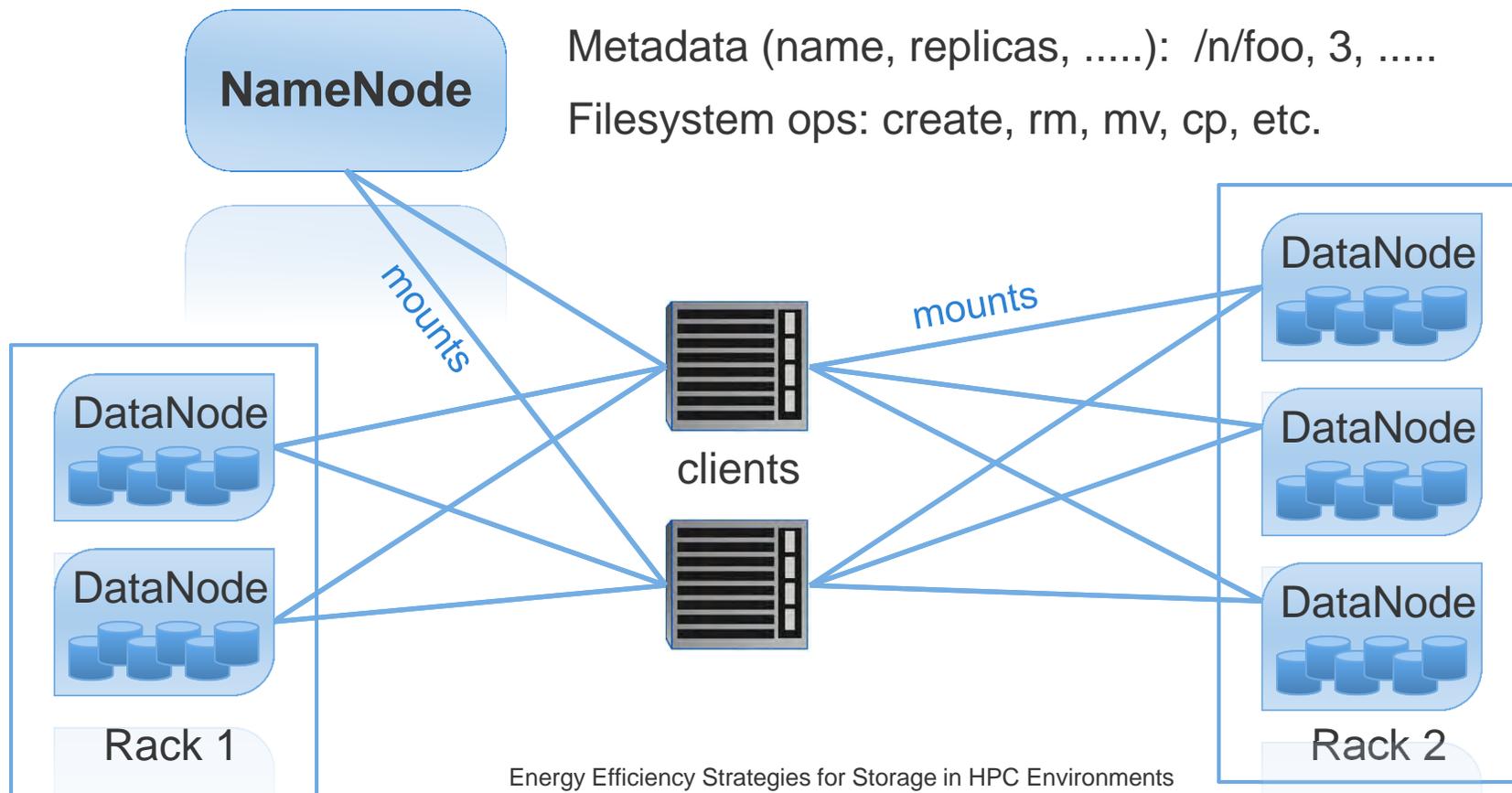
# Outline

- Typical Hadoop configuration
- Typical Lustre configuration
- NFS v4.1
- Replication – love and hate
- RAID 6
- Other technologies for increased efficiency

# Typical Hadoop configuration

## ➤ HDFS does the storage

- ◆ Master/slave architecture: single metadata server

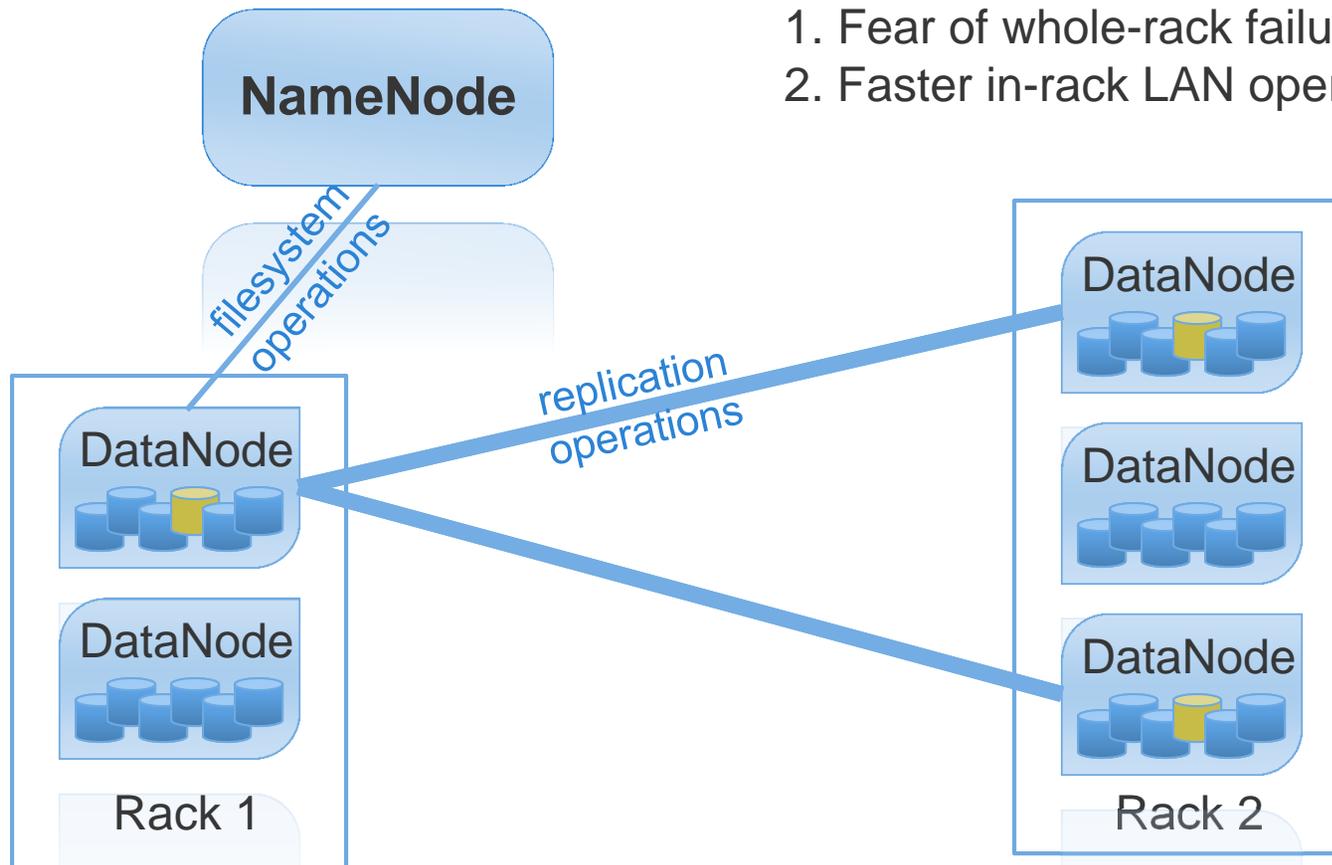


# Typical Hadoop configuration

## ► Replication

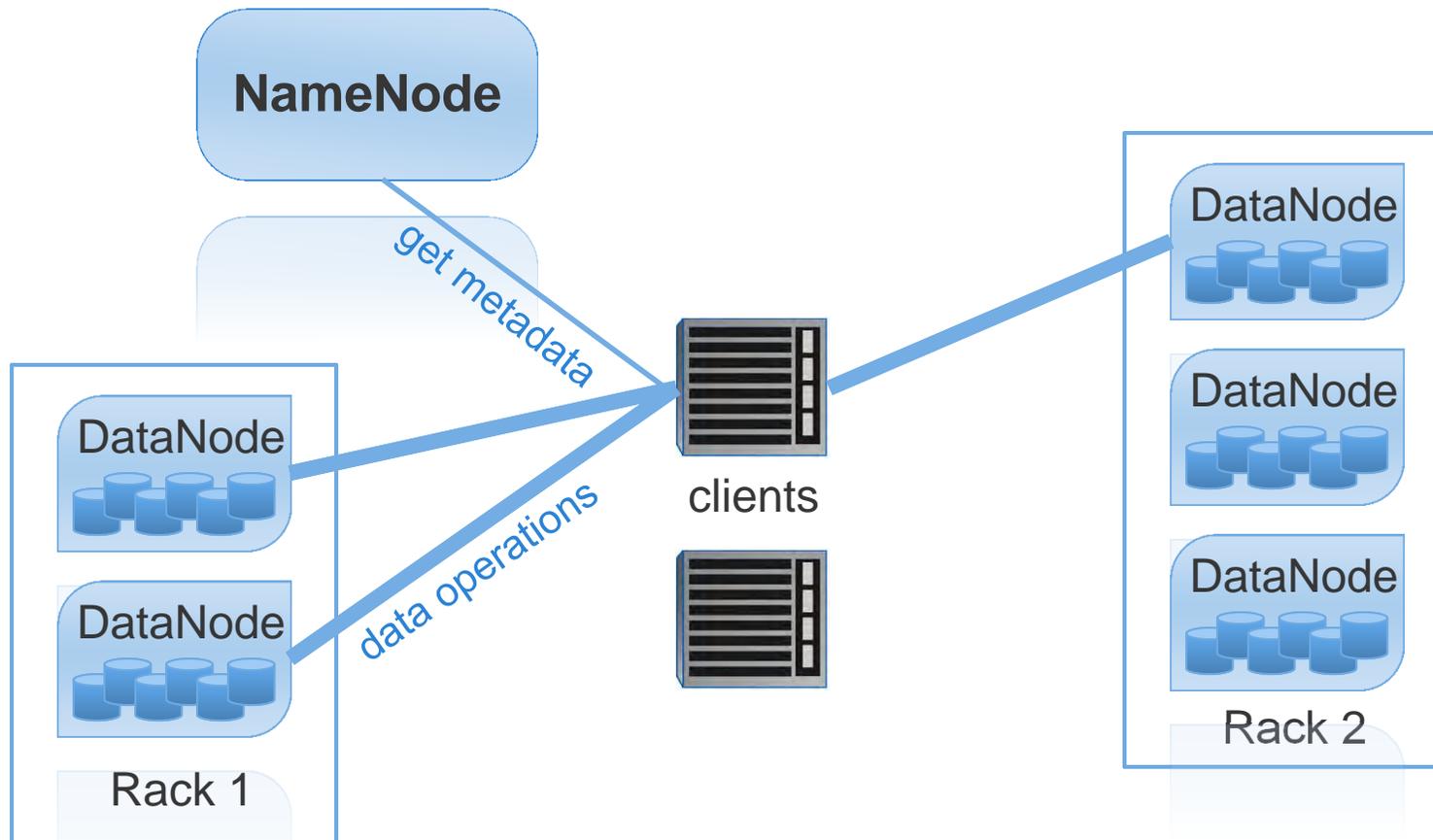
Design parameters:

1. Fear of whole-rack failure
2. Faster in-rack LAN operations



# Typical Hadoop configuration

## ➤ Typical data path operation

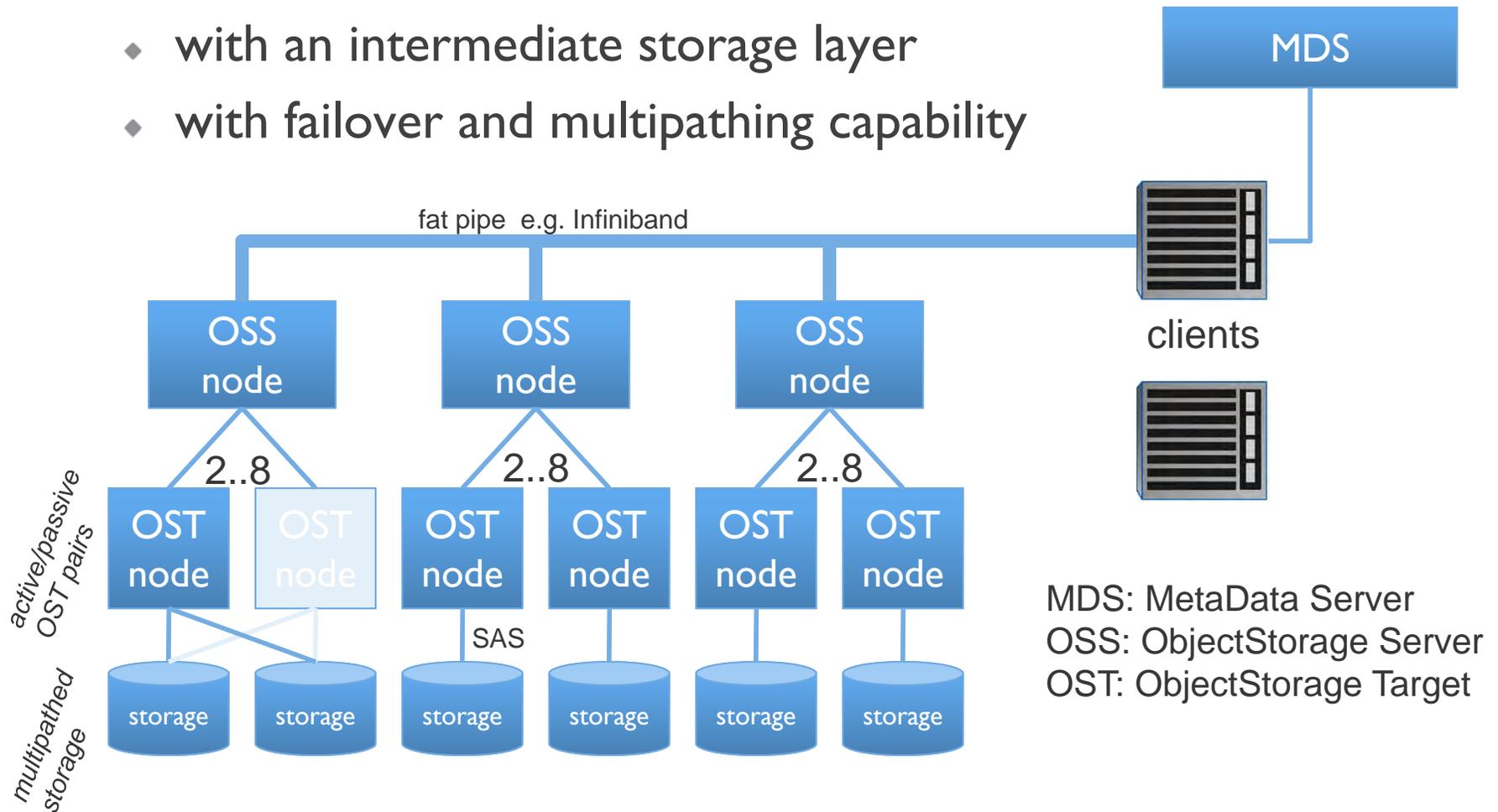


# Hadoop pros and cons (v 0.20)

- ▶ Parallel reads of large files
- ▶ Matching of computation to data location
  - ◆ Jobtracker → jobtasker communication
- ▶ “Doesn’t need RAID”
  - ◆ If “false is the new true”, this is a good thing
- ▶ Default replication value is 3 per data center
- ▶ Name server is a SPOF
  - ◆ Log-structured metadata store
  - ◆ Log replay only upon reboot ( 1/2 hour or more downtime)
  - ◆ Loss of NameNode filesystem is equivalent of a head crash
- ▶ No provision for high availability or disaster recovery

## ➤ Similar in concept to Hadoop

- ◆ with an intermediate storage layer
- ◆ with failover and multipathing capability



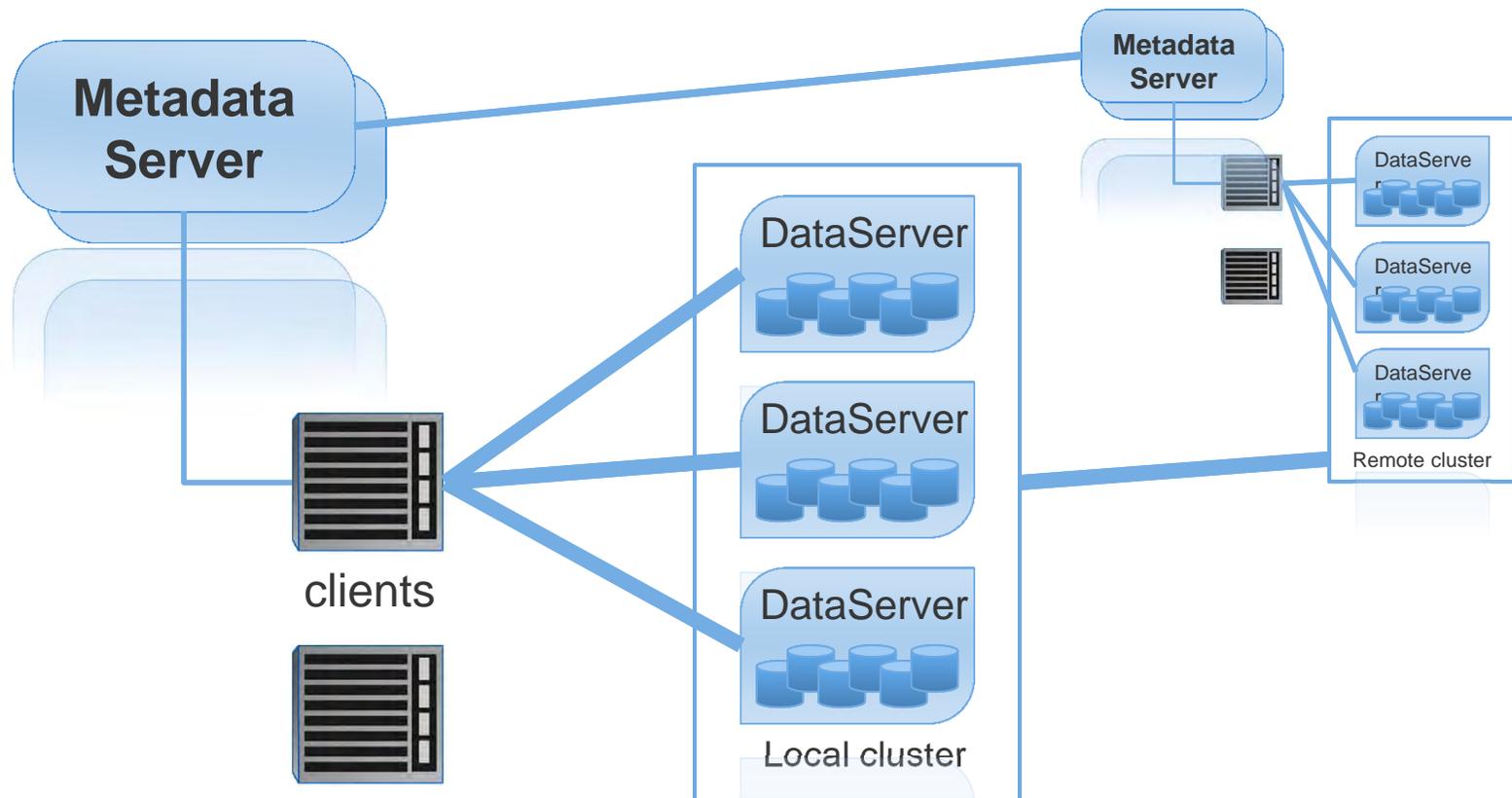
# Pros and cons of Lustre

- Much more “enterprise ready” than Hadoop
  - ◆ high availability
  - ◆ testing with and support for enterprise-ready Linux etc.
- Additional OSS layer insulates FS from rogue clients
  - ◆ safer
  - ◆ faster access to cached data
  - ◆ less efficient for streaming operations
- Less dependent on pure replication
- Active/passive failover
- MDS a SPOF

# NFS v4.1 (pNFS)

## ➤ Hadoop++, I mean ++++++

- ◆ Master/slave architecture : replicatable metadata server



# NFS v4.1 additional features

- **Delegations – similar to CIFS oplocks**
  - ◆ overlapping read lock support
  - ◆ cached local writes
- **Layouts – similar to Lustre layouts**
  - ◆ client-side control over striping etc.
- **Designed for enterprise storage**
  - ◆ high availability operations assumed and accommodated
  - ◆ enterprise level security incorporated
- **Allows sophisticated data management techniques**
  - ◆ e.g. single instance store, data dedup, delta snapshots, thin provisioning

# The theme so far (mostly)

➤ Lots of replication!

## ➤ Traditional data center system redundancy

- ◆ Overprovisioning – protect against volume-out-of-space application crashes
- ◆ Test/dev copies – protect live data from mutilation by unbaked code
- ◆ DR Mirror – protect against whole-site disasters
- ◆ Backups – protect against failures and unintentional deletions/changes
- ◆ Compliance archive – protect against heavy fines

## ➤ Big data systems like Hadoop

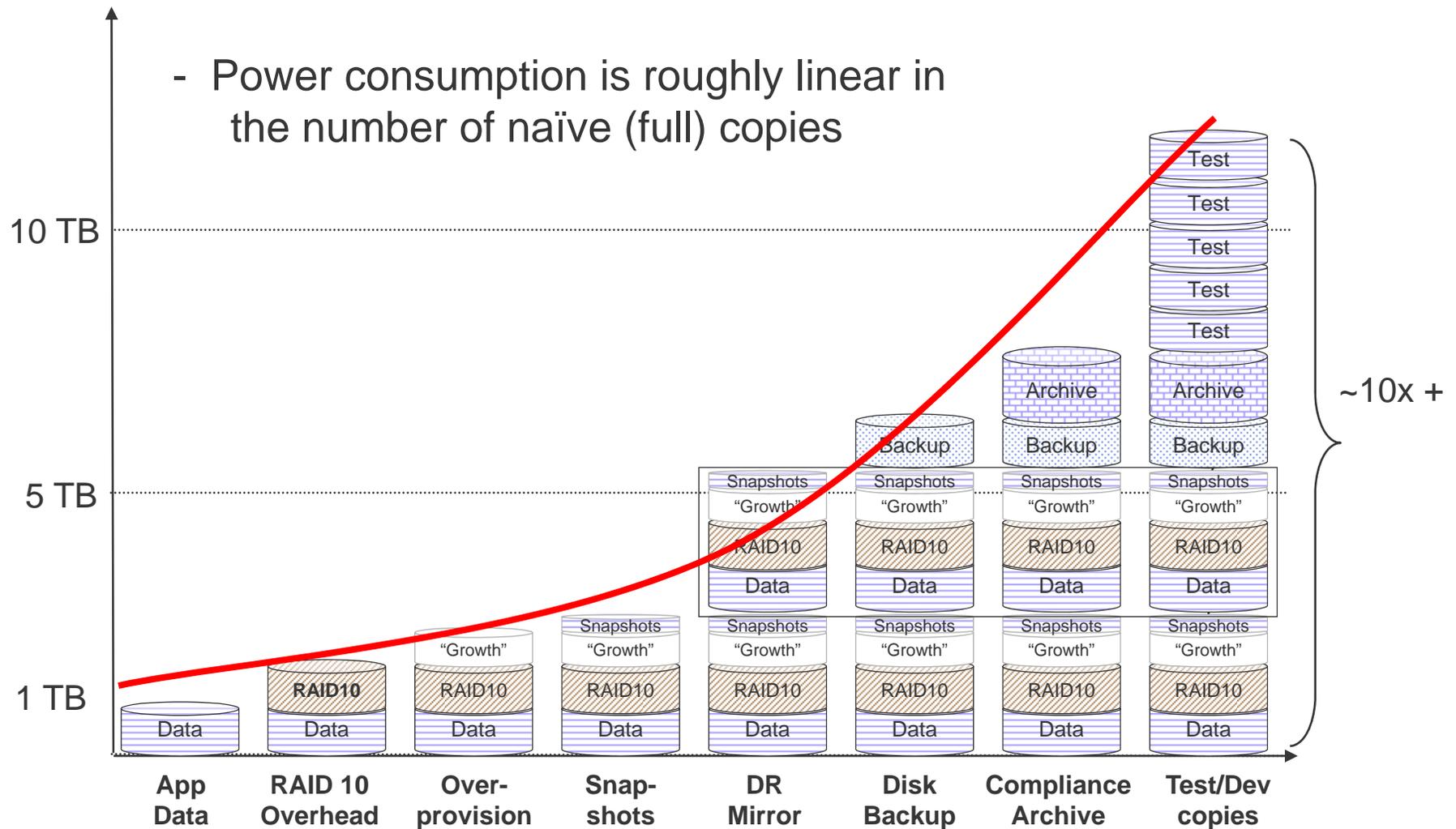
- ◆ Typically 3x replication *locally*
- ◆ Partly for performance

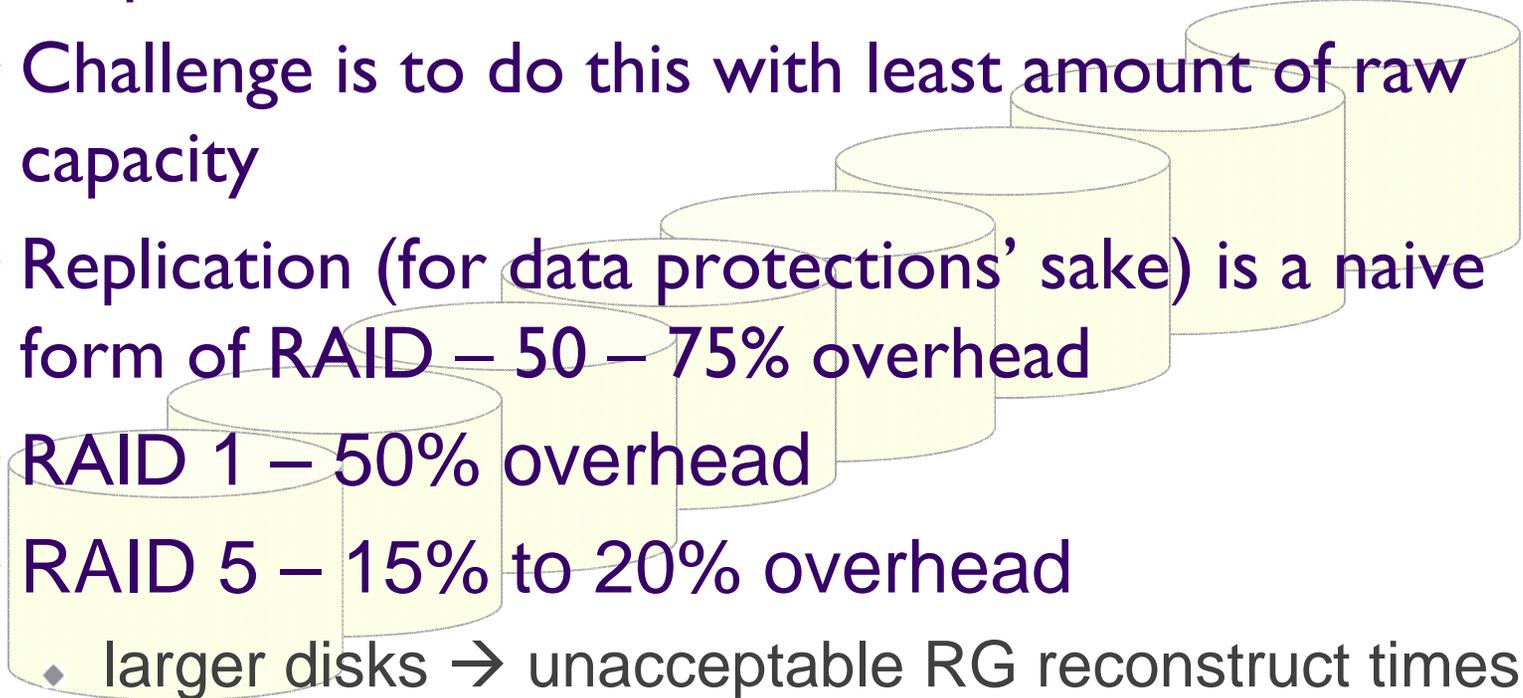
## ➤ “Brick” architectures

- ◆ Google, Microsoft
- ◆ At least 2x

shocking  
discovery: disk  
failure considered  
non-binary

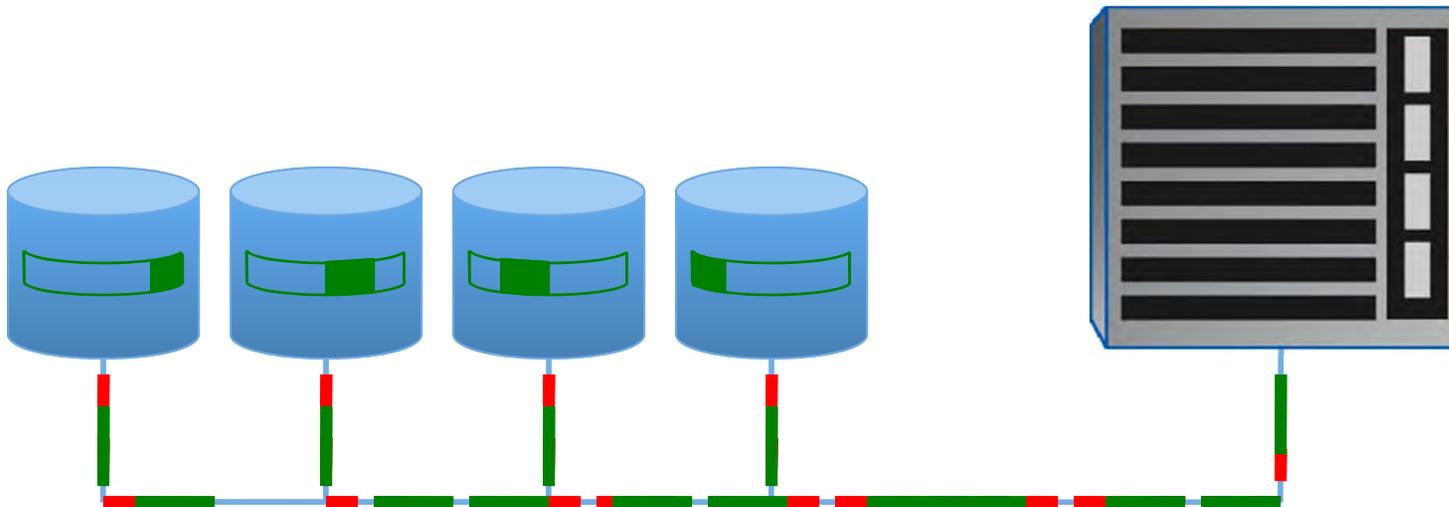
# Result of all that replication



- Requirement is to tolerate  $n$  failures
  - Challenge is to do this with least amount of raw capacity
  - Replication (for data protections' sake) is a naive form of RAID – 50 – 75% overhead
  - RAID 1 – 50% overhead
  - RAID 5 – 15% to 20% overhead
    - ◆ larger disks → unacceptable RG reconstruct times
  - RAID 6 – 15% to 25% overhead (8 + 2 common)
- 

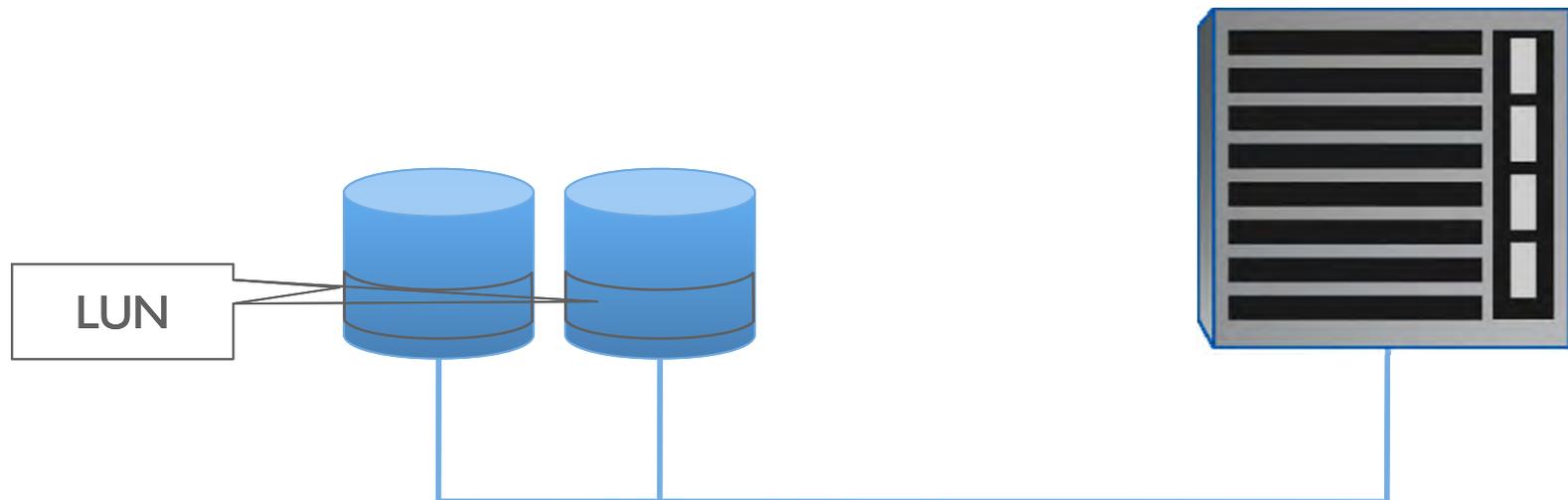
## ➤ RAID 0

- ◆ simple striping
- ◆ no parity
- ◆ performance gain



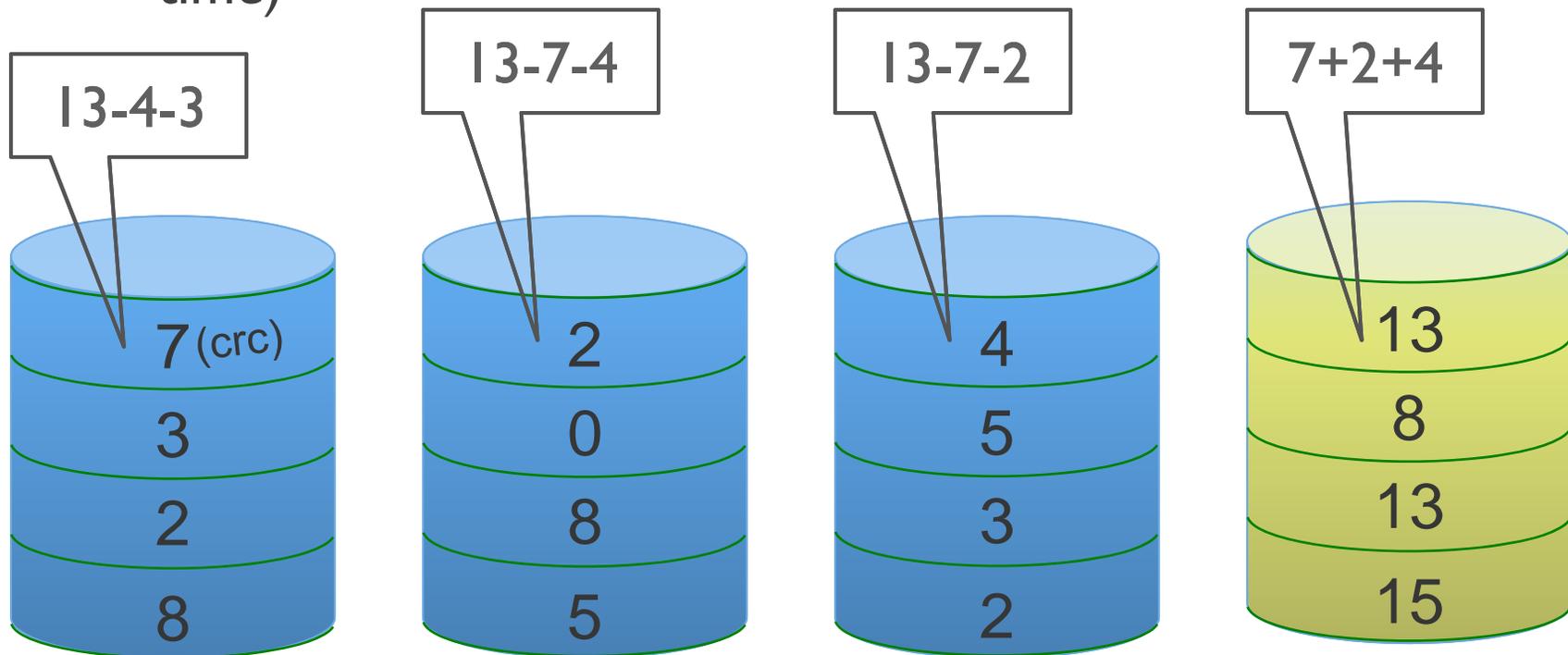
## ➤ Mirroring

- ◆ gold standard of data protection for many years
- ◆ simple duplication in hardware
- ◆ some potential performance benefit
- ◆ tolerates 1 failure in any 2-disk “RAID group”



## ➤ Parity on a separate spindle

- ◆ Parity is XOR of other stripes (equivalent to addition)
- ◆ Parity reconstructed by simple math (more disks → more time)



# Notes on RAID 4

- Parity disk is a hot spot in naive implementations
  - ◆ Needs aggressive write staging
- RAID group can be expanded with zeroed drives



# RAID 5

## ➤ Striped parity

- ◆ Same math – XOR
- ◆ Eliminates hot spot
- ◆ Some performance benefit touted
- ◆ Can't easily expand RAID group, however



# RAID 6

- Tolerates two drive failures per RAID group
- Row-diagonal parity (usually)
  - ◆ One parity stripe horizontally
  - ◆ One parity stripe diagonally
- Parity rebuild process is complex but provably correct
- Less than 3% performance hit in optimal implementations

# Notes on Gordon (SDSC)

- ◆ > 200 Tflops, 64TB DRAM, 320TB SSD in 64 nodes
- ◆ Expectations based on my experience:
  - ◆ High power use (> HDD per IOP and Gb/s)
  - ◆ Much faster for very large random I/O (WS > 64TB)
  - ◆ Faster for very large sequential reads
  - ◆ Slower for data ingest
  - ◆ Relatively small (< 1% the capacity of Sequoia)

# Flash and stash, tiering, MAID

## ➤ Flash and stash

- ◆ proven technology in enterprise space
- ◆ large flash caches (> 1TB) fronting low-power SATA disk
- ◆ resulting systems are hi-perf except on streaming writes
- ◆ expectation that flash will move to host over time
- ◆ recommend adding this technology to Lustre OSS nodes

## ➤ Tiering

- ◆ same idea, keep hot data on SSD, cold data on SATA
- ◆ runs well on Powerpoint

## ➤ MAID

- ◆ useful for hot spares

## Other energy efficiency technologies of note

- ◆ “Green” facility placement
- ◆ Water and natural cooling
- ◆ Hot aisle technologies
- ◆ Flywheel UPSes
- ◆ PUE monitoring
- ◆ Thin provisioning
- ◆ Compression
- ◆ Delta snapshots
- ◆ **Parity RAID**
- ◆ Deduplication and SIS
- ◆ Capacity vs. high performance drives
- ◆ ILM / HSM / Tiering
- ◆ MAID
- ◆ **SSDs / “Flash and stash”**
- ◆ Power supply and fan efficiencies

# NFS v4.1 – the future?

- FOSS development underway – UMich
- Support from major vendors
- Builds on 20 years of NFS development effort
- Fewer “moving parts” than Lustre
- Integrates well with other storage technologies
- Easier to distribute, archive
- Familiar POSIX interface for end users
- Easier to secure



# Additional material

- Tutorial: RAID
- How does a Lustre storage system work?
- How does a Hadoop cluster work?
- Get rid of COMs mostly
- San Diego flash system (Gordon)
- Tiering vs. Caching on flash
- Lustre vs NFSv4.1 vs MAID
- Wright, Disruptive Technologies for Data...

# Problem: making heat just to cool it

- Servers, storage and switches are HEATERS
  - ◆ 100% efficient energy-to-heat conversion
  - ◆ Rotating media uses 85% of max power *at idle!*
- A/C is a big “undo” mechanism for overheating
  - ◆ But less than 100% efficient (typically 70%)

> 60% of the power in  
a traditional data  
center does no IT  
work

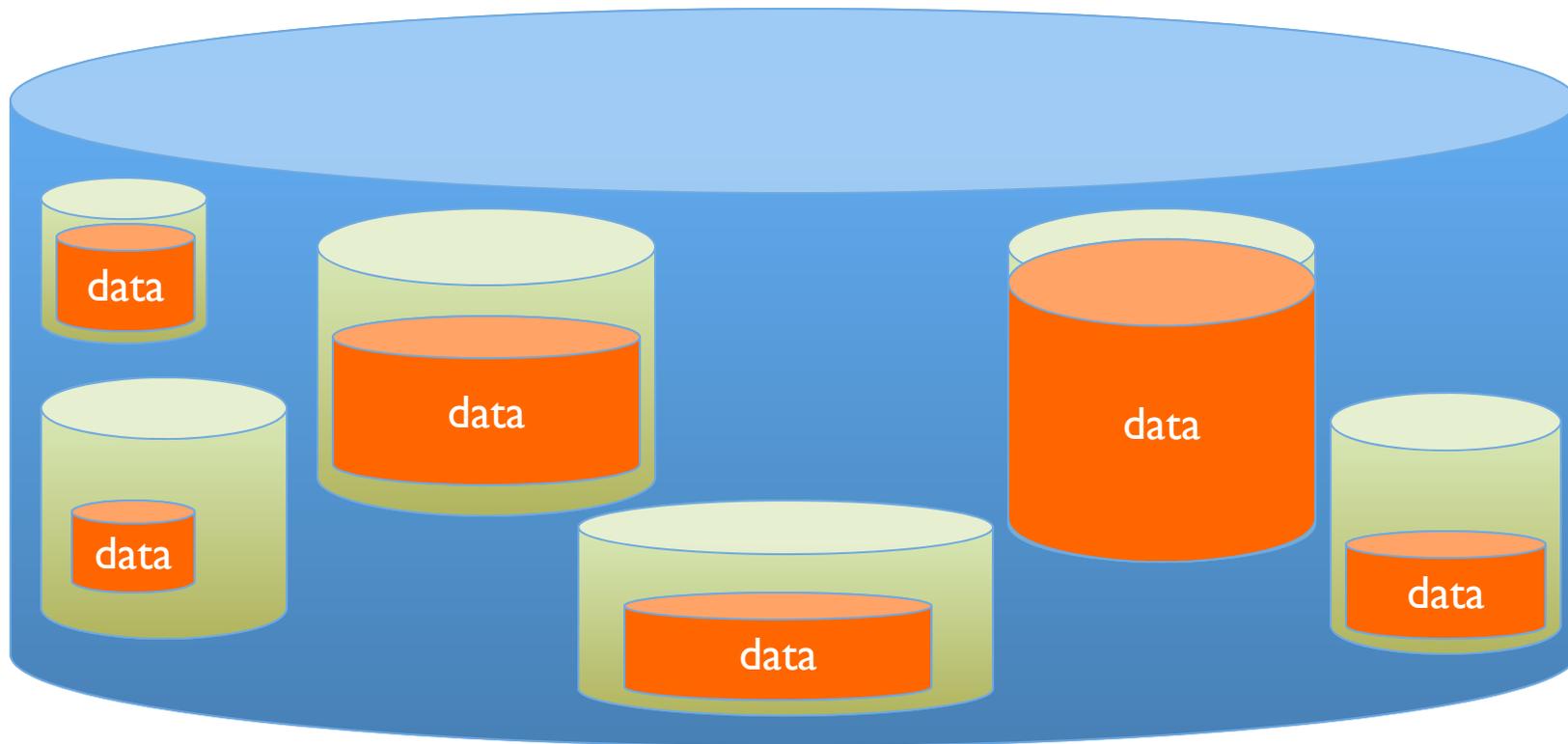
(PUE\* ~ 2.5)



\* PUE defined later

# Problem: unused space

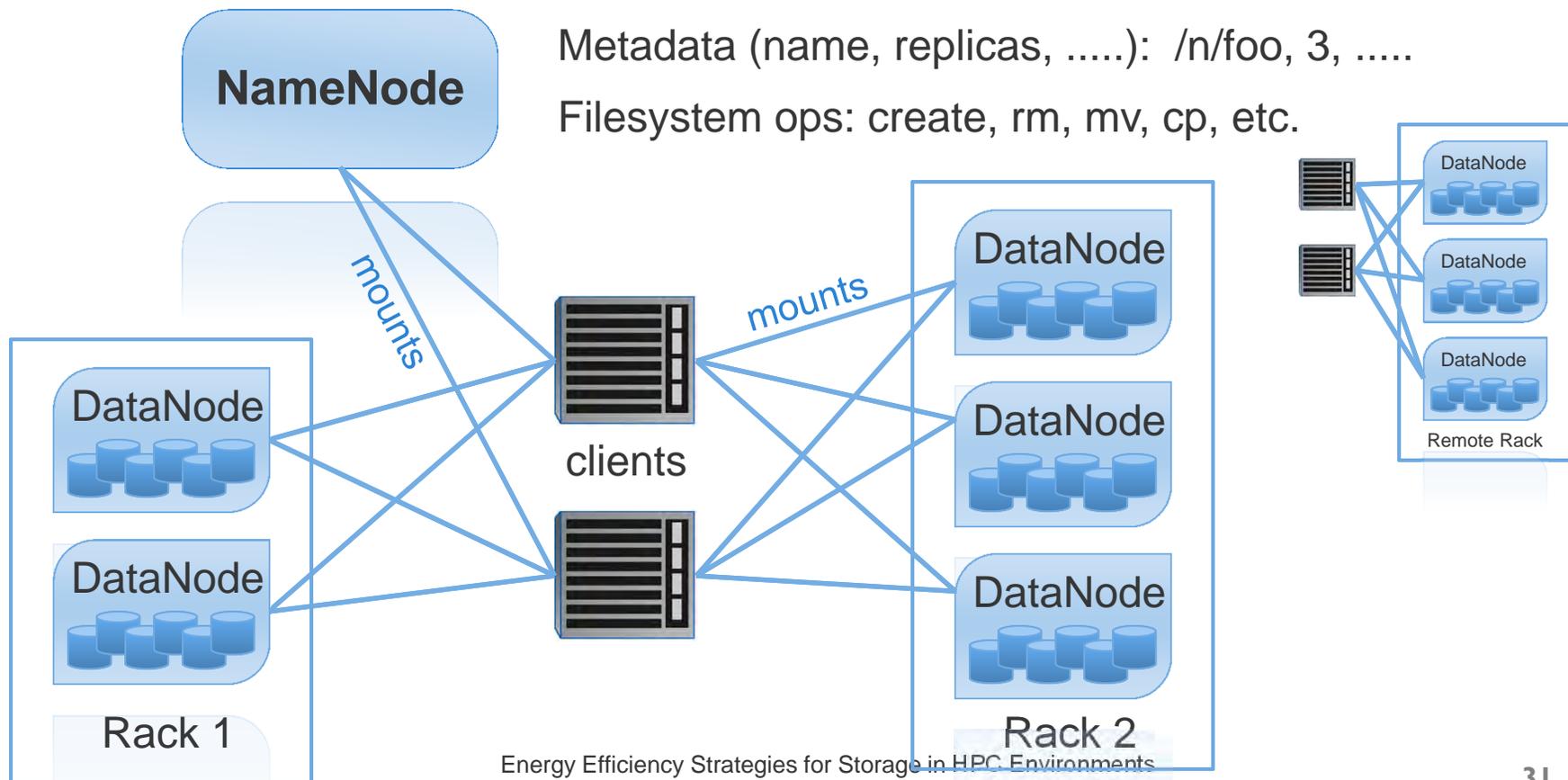
- Overprovisioning of systems
- Overprovisioning of containers



# Typical Hadoop configuration

## ► HDFS does the storage

- ◆ Master/slave architecture: single metadata server



# Oak Ridge Labs “Spider”

Aggregate Bandwidth	240GB/s
Storage Systems	48 x DDN S2A9900 Storage Arrays
Hard Drives	13,440 1TB SATA Hard Drives
Aggregate Capacity	13.44 Petabytes (Raw), 10.7 Petabytes (Usable – 8+2 RAID 6)
Lustre Storage Servers	192 Lustre OSS Servers
Cabling	Over 1,000 20Gb InfiniBand Cables
Data Center Cabinets	32 Data Center Racks, 572 ft

## ➤ PUE – Power Use Efficiency

$$PUE = \frac{Total\_Facility\_Power}{IT\_Power}$$

## ➤ Weighted upward by

- ◆ UPS and power conditioning inefficiencies
- ◆ **Inefficient cooling**

## ➤ Traditionally 2.5, modern best practice = 1.25

## ➤ Can be gamed

- ◆ Use of equipment fans to drive hot air exhaust

# Hot aisle / cold aisle technologies

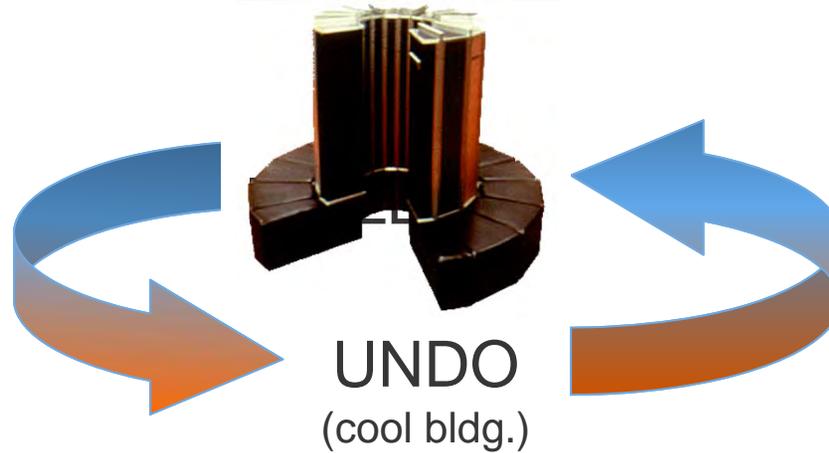
- Segregate airflows into hot and/or cold aisles (backs and fronts of servers)
  - ◆ More precise control
  - ◆ Allows higher temperature differentials (more efficient)
  - ◆ Current trend toward hot aisle containment with cold air plenum
  - ◆ Must-have: blanking plates
    - > Very important
  - ◆ Normally deployed in comb. w/ air economizers



# Green facilities

e.g.

BEFORE



AFTER



**40% SAVINGS (w/ air economizer)  
(PUE: 2.5 → 1.5)**

# Contributing to a green facility

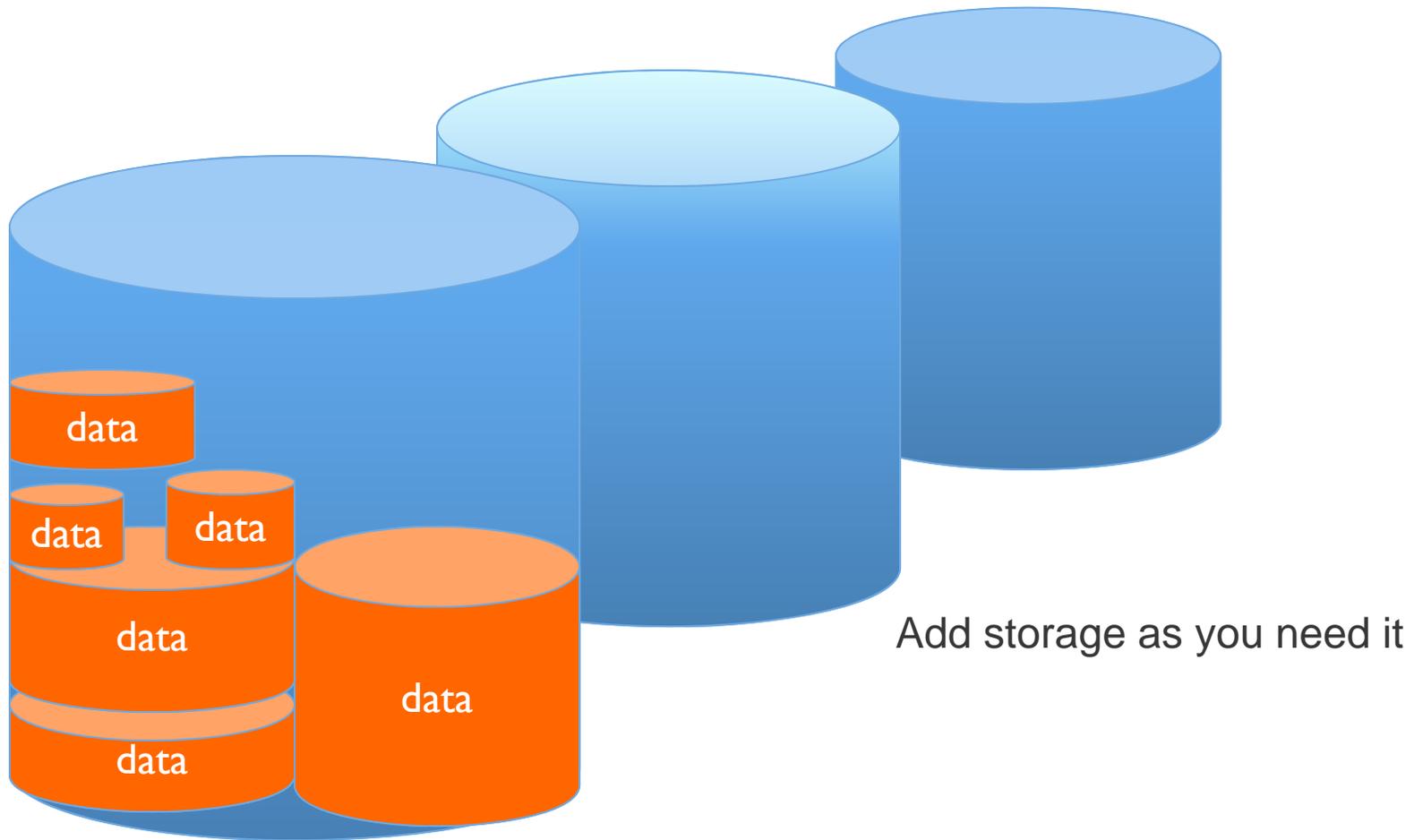
## ➤ Hot aisle containment

- ◆ Need temperature monitoring
  - › Recommend rack level PDUs
- ◆ Rigorous use of blanking plates required
- ◆ With appropriate air economizers etc., the single biggest contribution to energy conservation you can make

## ➤ Note re “green power”

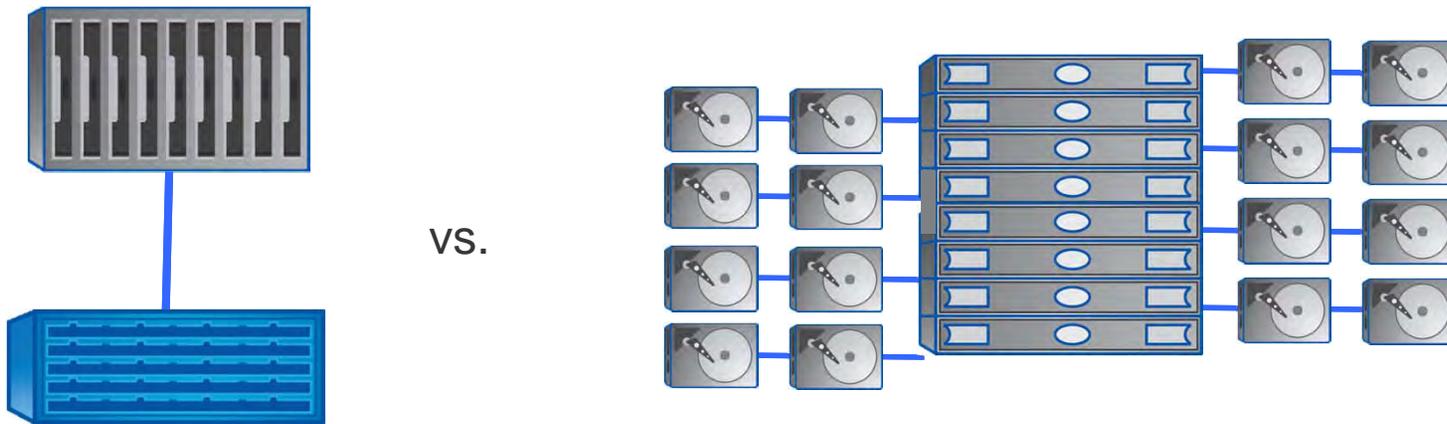
- ◆ It's only green if you generate it yourself (TGG)

# Thin provisioning

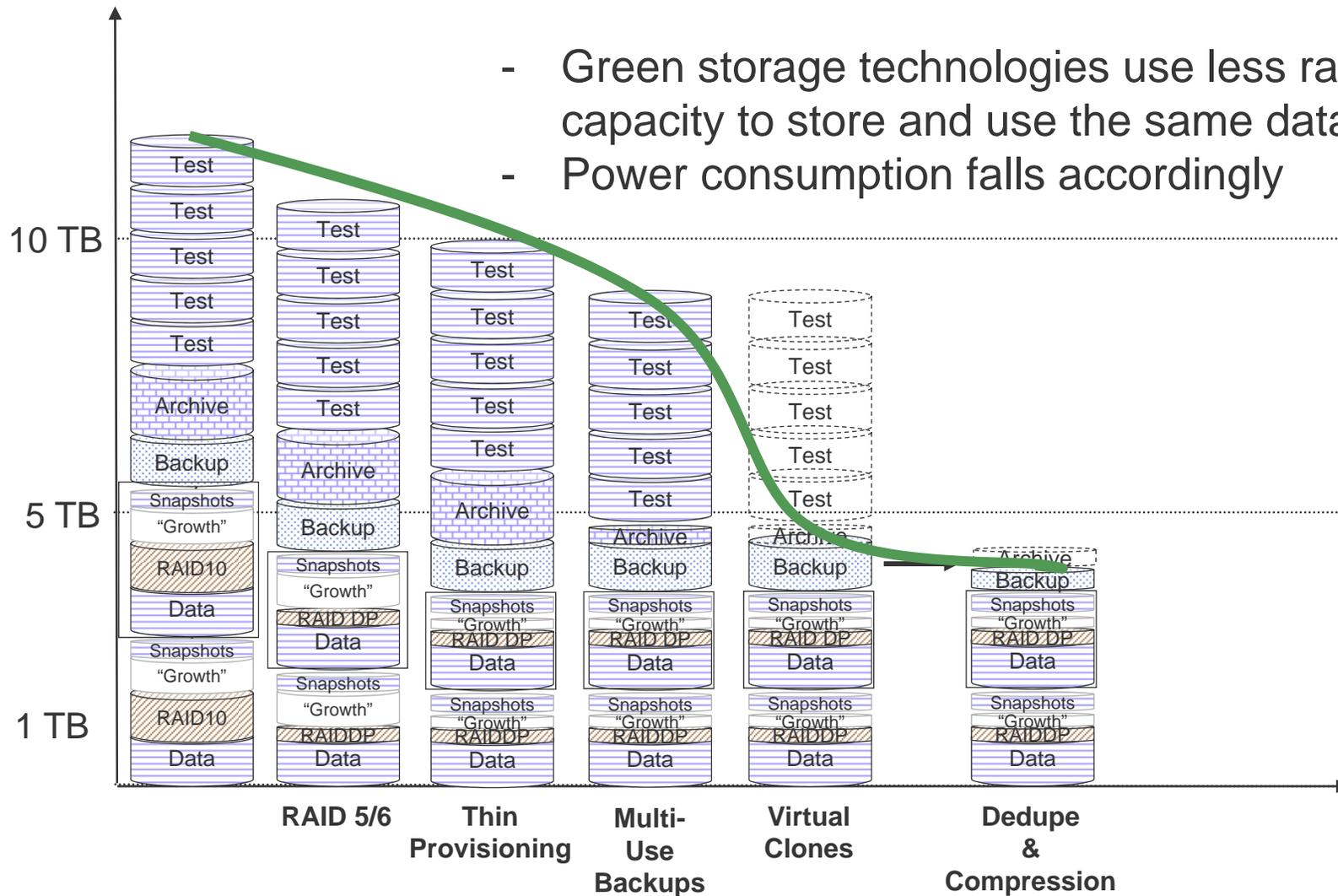


# Thin provisioning notes

- Most useful w/ variable ratios of compute and storage requirement
  - ◆ Especially true in cloud PaaS and SaaS environments
  - ◆ TP biases toward centralized storage



# Solution: Capacity optimization technologies



- Green storage technologies use less raw capacity to store and use the same data set
- Power consumption falls accordingly

# Green software technologies

- Compression
- Delta snapshots
- Thin provisioning
- Parity RAID
- Deduplication and SIS



## ➤ Compression

- ◆ Old and venerable
- ◆ Almost a “gimme”
- ◆ Configuration matters
  - › Compress before encrypting, decrypt before decompressing

## ➤ Data sharing

- ◆ Form of deduplication
- ◆ Data in snapshot shared with live data until one of them is written
- ◆ Two fundamental techniques
  - › Copy Out on Write
  - › Write to new live location

## ➤ Mainly useful in HPC for VM booting

- ◆ One storage system can support a couple hundred diskless servers

# Deduplication and SIS

- Find duplicates at some level, substitute pointers to a single shared copy
- Block or sub-file based (dedup)
- Content or name based (SIS \*, “file folding”)
- Inline (streaming) and post-process techniques
- Savings increase with number of copies found
- Performance hit may be too expensive for HPC

\* SIS = Single Instance Store



**Check out the SNIA  
Dictionary !**

**[www.snia.org/dictionary](http://www.snia.org/dictionary)**

# SSDs (Solid State Disks)

## ➤ Pros

- ◆ Great READ performance
- ◆ At rest power consumption = 0
- ◆ No access time penalty when idle (cf. MAID)
- ◆ No need to keep some disks spinning (cf. MAID)

## ➤ Cons

- ◆ WRITE performance usually < mechanical disks
- ◆ Cost >> mechanical disks except at very high perf points
- ◆ Wear leveling requires a high space overhead

## ➤ Note: these dynamics changing rapidly with time

# “Flash and stash”

- ▶ Large arrays of SATA-based disks fronted by large flash caches
  - ◆ > 1TB flash
  - ◆ Great for high I/O, esp. w/ contained working sets
  - ◆ Reduced power (SATA vs. SAS)
  - ◆ Not useful for write-intensive workloads

# Power supply and fan efficiencies

- ▶ Efficiency of power supply an up front waste
  - ◆ Formerly 60-70%
  - ◆ Nowadays 80-95%
    - › Climate Savers
    - › 80plus group
- ▶ Variable speed fans
  - ◆ Common nowadays
  - ◆ Software (OS) control

- ▶ Many HPC workloads favor centralized storage and management
  - ◆ nonproliferation, counter-terrorism, energy security, etc.
  - ◆ health care analytics (SOX, HIPAA)
  - ◆ pharma research (FDA)
  - ◆ weapons
- ▶ Security and retention/compliance
  - ◆ both easier and better with the enforcement point in the storage layer
  - ◆ physical security still necessary

# Other points

## ➤ Centralized storage also offers

- ◆ MUCH better data management
- ◆ Very efficient DR and remote replication
- ◆ Strong vendor support
- ◆ Great virtualization support

## ➤ But

- ◆ capabilities may be wasted in extreme HPC environments
- ◆ increased up-front cost often a psychological barrier

# Key takeaways

- Hot aisle containment
  - RAID 6 is current best practice
  - Headless server configurations are possible
  - Use of software capacity optimizations highly recommended in cloud-style environments
- 
- For more detail:  
<http://www.snia.org/education/tutorials/2009/fall#green>